Methods 129 (2017) 81–88

Contents lists available at ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth

Drug-target interaction prediction using ensemble learning and dimensionality reduction

Ali Ezzat [a], Min Wu [b], Xiao-Li Li [b,*], Chee-Keong Kwoh [a]

[a] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[b] Institute for Infocomm Research (I^2R), A*Star, Singapore

ARTICLE INFO

Article history:
Received 16 January 2017
Received in revised form 3 April 2017
Accepted 18 May 2017
Available online 24 May 2017

Keywords:
Drug-target interaction prediction
Feature subspacing
Dimensionality reduction
Ensemble learning
Kernel ridge regression

ABSTRACT

Experimental prediction of drug-target interactions is expensive, time-consuming and tedious. Fortunately, computational methods help narrow down the search space for interaction candidates to be further examined via wet-lab techniques. Nowadays, the number of attributes/features for drugs and targets, as well as the amount of their interactions, are increasing, making these computational methods inefficient or occasionally prohibitive. This motivates us to derive a reduced feature set for prediction. In addition, since ensemble learning techniques are widely used to improve the classification performance, it is also worthwhile to design an ensemble learning framework to enhance the performance for drug-target interaction prediction.

In this paper, we propose a framework for drug-target interaction prediction leveraging both feature dimensionality reduction and ensemble learning. First, we conducted feature subspacing to inject diversity into the classifier ensemble. Second, we applied three different dimensionality reduction methods to the subspaced features. Third, we trained homogeneous base learners with the reduced features and then aggregated their scores to derive the final predictions. For base learners, we selected two classifiers, namely *Decision Tree* and *Kernel Ridge Regression*, resulting in two variants of ensemble models, *EnsemDT* and *EnsemKRR*, respectively.

In our experiments, we utilized AUC (Area under ROC Curve) as an evaluation metric. We compared our proposed methods with various state-of-the-art methods under 5-fold cross validation. Experimental results showed *EnsemKRR* achieving the highest AUC (94.3%) for predicting drug-target interactions. In addition, dimensionality reduction helped improve the performance of *EnsemDT*. In conclusion, our proposed methods produced significant improvements for drug-target interaction prediction.

1. Introduction

Determining drug-target interactions is very important for drug discovery in pharmaceutical science. For example, drug repositioning, which is a rising trend in drug discovery, focuses on discovering the interactions between new targets and existing drugs [1]. However, wet-lab experiments to detect these interactions are usually costly, time-consuming and labor-intensive [2]. It is thus highly motivated to develop computational methods to predict drug-target interactions, which can effectively narrow down the search space of the candidates to be investigated by wet-lab techniques, to reduce the cost and effort involved. Nowadays, computational methods have become prevalent due to the availability of large online databases that store information on known drug-target interactions such as KEGG [3], DrugBank [4], ChEMBL [5] and STITCH [6].

Currently, efforts for predicting drug-target interactions (either *novel drug discovery* efforts or *drug repositioning* efforts for existing drugs) are being supported by different computational methods working on different types of data sources [1,7]. We can divide these methods into three categories, namely, *ligand-based* approaches, *docking-based* approaches and *chemogenomic* approaches. Firstly, *ligand-based* approaches predict drug target interactions based on the similarity between the target proteins' ligands. Secondly, *docking-based* approaches [8,9] utilize 3D structure information of a target protein and then run simulations to estimate the likelihood that it will interact with a certain drug based on their binding affinity and strength. Finally, *chemogenomic* approaches [10] usually leverage the chemical and genomic information from drugs and targets, respectively, as well as the known existing drug-target interactions for predictions.

* Corresponding author.
E-mail address: xlli@i2r.a-star.edu.sg (X.-L. Li).

http://dx.doi.org/10.1016/j.ymeth.2017.05.016
1046-2023/

The applicability of the approaches in the first and second categories is often limited due to the lack of ligands and 3D structures available, respectively, for some target proteins. Therefore, *chemogenomic* approaches became more popular for predicting drug-target interactions. *Chemogenomic* approaches model the task of predicting drug-target interactions as a machine learning problem. They take the data on known interactions along with the properties of the drugs and targets involved to train a classifier and subsequently predict novel interactions by the trained classifier. According to a recent survey paper [1], different machine learning techniques can be either categorized as *feature-based methods* [11–13] or *similarity-based methods*. Similarity-based methods include kernel-based methods [14–16], matrix factorization [17–19] and graph-based methods [20,21].

Our work has two major motivations. Firstly, ensemble learning techniques that aim to integrate multiple base classifiers/learners for robust and accurate predictions have been widely used to improve the classification performance. It was thus imperative for us to design ensemble learning models to further improve the prediction performance of drug-target interactions. Secondly, we observe that the information for drugs and targets as well as the number of known drug-target interactions keep on increasing. The above machine learning models, especially our proposed ensemble learning models, with a large number of base learners, would become inefficient for the prediction tasks. To address this issue, we propose to apply different types of dimensionality reduction techniques to remove those noisy, redundant, or irrelevant information so that it can effectively reduce the data size to a manageable level before we build our machine learning models.

In this paper, we propose a framework for drug-target interaction prediction that uses ensemble learning and feature dimensionality reduction. Firstly, we conducted feature subspacing to inject diversity for classifier ensemble. Secondly, we investigated three different dimensionality reduction methods, namely *Singular Value Decomposition*, *Partial Least Squares* and *Laplacian Eigenmaps*, to the subspaced features (obtained by feature subspacing) to reduce the number of features. Finally, we trained homogeneous base learners with the reduced features and then derived the final predictions by aggregating their scores. In general, any classification method can be applied as a base learner in our framework. Particularly, in our work, we selected two efficient base learners, namely *Decision Tree* [22] and *Kernel Ridge Regression* [23], and presented two variants of ensemble models, denoted as *EnsemDT* and *EnsemKRR*, respectively. Experimental results demonstrated that *EnsemDT* outperforms existing feature-based methods (e.g. SVM and Random Forest), and *EnsemKRR* achieves the best performance (e.g. the highest AUC 0.943) for drug-target interaction prediction, indicating that our proposed methods are potentially useful for practical drug discovery.

2. Related work

In this section, we present a brief review for the state-of-the-art methods for drug-target interaction prediction. Note that they are existing competing chemogenomic methods that we will compare our proposed methods against later in the experiments. In a recent survey paper [24], these competing methods can be further divided into two categories: *feature-based* methods and *similarity-based* methods. The difference between them is that they take input data in different representations as explained below.

2.1. Feature-based methods

Feature-based methods take their inputs in the form of feature vectors, representing a set of instances (i.e. drug-target pairs) along with their corresponding class labels (i.e. binary values indicating whether or not an interaction exists). *Decision Tree* (*DT*), *Random Forest* (*RF*) [25] and *Support Vector Machines* (*SVM*) [26] are typical feature-based methods to build classification models based on the labeled feature vectors. In particular, *RF* and *SVM* [26] are applied for predicting drug-target interactions in [11]. In these machine learning methods, known interactions represent positive instances and non-interactions denote negative instances. To be more precise, negative instances here include both non-interactions and unknown drug-target interactions (false negatives). Due to the huge number of possible negative instances, to balance the positive and negative training data, feature-based methods usually undersample the negative data prior to training and prediction; i.e. negative instances are randomly sampled to have the same size as the positive instances.

2.2. Similarity-based methods

Similarity-based methods, on the other hand, take their inputs explicitly in the form of two similarity matrices for the drugs and targets, respectively, along with an interaction matrix that indicates which pairs of drugs and targets interact. A number of similarity-based methods have been proposed, including *Nearest Profile* (*NP*), *Weighted Profile* (*WP*) [10], *Network-based Inference* (*NBI*) [21], *RLS-avg* [15] and *RLS-kron* [27] (where *RLS* is short for *Regularized Least Squares*).

WP predicts an interaction profile for new drug by performing a weighted average of the other drugs' profiles, the weights being the similarities between the new drug and the other drugs. *NP* is similar to *WP* except that it predicts an interaction profile for a new drug using only its nearest neighbor (i.e. the drug most similar to it).

NBI models the prediction problem as a network where the drugs and targets are represented as nodes, and the interacting drug-target pairs are connected by edges. The network diffusion technique is then applied to propagate interaction information throughout the drug-target interaction network.

RLS-avg and *RLS-kron* are both techniques that are based on *Kernel Ridge Regression* [26], where *Least Squares* operates in the space induced by the kernels used. Each of them works on the kernels in a different manner for training and prediction. *RLS-avg* uses kernel ridge regression along with the concept of bipartite local models for prediction. In other words, a classifier is trained for each drug using only the target kernel K^t, while for each target, a classifier is trained using only the drug kernel K^d. After that, a final prediction score is then given for each drug-target pair by averaging the results from both the drug and target side. *RLS-kron*, on the other hand, first takes the Kronecker product of K^d and K^t and then builds a single classifier for prediction.

3. Materials and methods

3.1. Data

In this work, we use two datasets that have been used in previous work. The first dataset is from [13] which includes the drug-target interaction data as well as the features used to represent drugs and targets. Statistics for this dataset are shown in Table 1 below. The features of the drugs and targets have been calculated using the *Rcpi* [28] package and the *PROFEAT* [29] web server, respectively. Examples of drug features used include constitu-

Table 1
Statistics of Dataset 1.

Drugs	Targets	Interactions
5,877	3,348	12,674

Table 2
Statistics of Dataset 2.

Drugs	Targets	Interactions
1,862	1,554	4,809

tional, topological and geometrical descriptors, while target features include amino acid composition, pseudo-amino acid composition and CTD (composition, transition, distribution) descriptors among others. In addition, the features have been properly normalized before they are used in all the machine learning methods in this work.

The second dataset is from [30]. Statistics for this dataset are shown in Table 2 below. Drugs in this dataset are represented as PubChem fingerprints (i.e. binary vectors where each element indicates the presence or absence of one of 881 known chemical substructures). On the other hand, targets are represented as fingerprints that indicate the presence or absence of 876 different protein domains that are obtained from the Pfam database [31].

For notation, let $Y \in \mathbb{R}^{n\times m}$ be the drug-target interaction matrix having n drug rows and m target columns where $Y_{ij} = 1$ if drug d_i and target t_j interact and 0 otherwise. In addition, let $D \in \mathbb{R}^{n\times p}$ and $T \in \mathbb{R}^{m\times q}$ be the feature matrices for the drugs and targets, respectively, where p and q are the numbers of drug and target features, respectively.

3.2. Dimensionality reduction

Dimensionality reduction projects our training data into a feature space with a lower dimension, and thus reduces the memory requirement and decreases the time complexity for feature-based methods. Next, we briefly introduce three different dimensionality reduction techniques to be investigated in this paper, namely *Singular Value Decomposition*, *Partial Least Squares* and *Laplacian Eigenmaps*. Note that we have used k below ($k \ll p$, $k \ll q$) as an adjustable parameter for controlling the dimensionality of the reduced feature space obtained from these techniques.

3.2.1. Singular Value Decomposition (SVD)

In our work, we use *Truncated SVD* which is an economical variant of standard *SVD*. It takes a given matrix $A \in \mathbb{R}^{n\times p}$ and decomposes it into $U \in \mathbb{R}^{n\times k}$, $S \in \mathbb{R}^{k\times k}$ and $V \in \mathbb{R}^{p\times k}$ such that $A = USV^\top$. S is a diagonal matrix containing the largest k eigenvalues of A. The reduced matrix can then be obtained as $\widetilde{A} = US$. This procedure is applied to both D and T to obtain $\widetilde{D} \in \mathbb{R}^{n\times k}$ and $\widetilde{T} \in \mathbb{R}^{m\times k}$. For the remainder of this paper, we refer to *Truncated SVD* as *SVD*.

3.2.2. Partial Least Squares (PLS)

PLS [32] is a supervised dimensionality reduction technique that takes a matrix of predictor variables $A \in \mathbb{R}^{n\times p}$ and a response matrix $B \in \mathbb{R}^{n\times m}$ as input. After first centering A and B (by subtracting off column means to get centered variables), *PLS* then builds the model

$$A = UP^\top + E$$
$$B = VQ^\top + F$$

where $U \in \mathbb{R}^{n\times k}$ and $V \in \mathbb{R}^{n\times k}$ are projections of A and B, respectively. Here, P and Q are loading matrices, and E and F are error terms that are assumed to be i.i.d. random normal variables. The model implicitly aims to maximize the covariance between U and V. We reduce the dimensionality of the drug matrix D by replacing A and B with D and Y in the above equation, respectively. The target matrix T can be processed in a similar manner by replacing A and B with T and $Y^\top$, respectively.

3.2.3. Laplacian Eigenmaps (LapEig)

Laplacian Eigenmaps [33] is a manifold learning algorithm which performs non-linear dimensionality reduction. Given a feature matrix $X \in \mathbb{R}^{n\times p}$, *Laplacian Eigenmaps* constructs a t-nearest-neighbor graph W where $W_{ij} = \exp(-\|x_i - x_j\|^2)$ if x_i and x_j are neighbors and 0 otherwise. It then finds a k-dimensional representation of X, $Z \in \mathbb{R}^{n\times k}$, by minimizing the following objective function,

$$Z = \underset{Z'}{\operatorname{argmin}} \sum_{ij} W_{ij} \|z'_i - z'_j\|^2,$$

where neighboring observations z'_i and z'_j are encouraged to be similar in the transformed space if their corresponding x_i and x_j are also similar to each other in the original feature space as indicated by W_{ij}.

3.3. Our proposed ensemble methods

In this section, we present the details for our two proposed ensemble methods, namely *EnsemDT* and *EnsemKRR*.

3.3.1. Ensemble of Decision Trees with Different Negative Sets (EnsemDT)

Algorithm 1 presents an overview for our ensemble learning method *EnsemDT* that employs *Decision Tree* as the base learner. For each base learner, the minority class instances (or the positive instances) are included in the training set along with a randomly sampled set of negative instances from the majority class. The number of negative instances sampled per base learner is controlled by the parameter, *npRatio* (short for *negative-to-positive ratio*). For example, if *npRatio* = 5, then the number of sampled negative instances is 5 times the number of positive instances in the training set. After that, feature subspacing is performed (i.e. a different subset of features is randomly selected per base learner) to inject diversity into the ensemble – diversity is generally known to be beneficial to the prediction performance of ensembles [22]. This is then followed by dimensionality reduction to reduce the size of the data; that is, using a parameter r (where $r \leqslant 1$), a subset of $r \times |F|$ features are randomly sampled for each base learner (where F is the full feature set). After training and predicting with M base learners, the results are aggregated by simple averaging.

Algorithm 1: *EnsemDT - Decision Tree Ensemble*

Input: P = positive instance set,
N = negative instance set,
D = feature matrix for the drugs,
T = feature matrix for the targets,
$npRatio$ = *negative-to-positive ratio*,
r = dimensionality reduction parameter,
M = the number of trees in the ensemble

Result: $ensemble$ = trained ensemble

begin
  for $i \leftarrow 1$ ***to*** M **do**
    Randomly sample $N_i \in N$ until $|N_i| = npRatio \times |P|$
    $TrainingSet = P \cup N_i$
    D_i = randomly selected feature subset (of size $r \times p$)
    $D_i = DimRed(D_i)$ //dimensionality reduction
    T_i = randomly selected feature subset (of size $r \times q$)
    $T_i = DimRed(T_i)$ //dimensionality reduction
    $TrainingSet = TrainingSet(D_i, T_i)$ //form instance vectors
    $tree_i$ = train a decision tree model using $TrainingSet$
  return $ensemble = \frac{1}{M}\sum_{i=1}^{M} tree_i$

Note that we derive the feature vectors for the instances (i.e. drug-target pairs) by concatenating the feature vectors of the involved drugs and targets. For example, an instance (d_i, t_j) would have a feature vector $[D_{i,*}, T_{j,*}]$, where $D_{i,*}$ and $T_{j,*}$ are the row vectors in D and T that correspond to drug d_i and target t_j, respectively. Each base learner (i.e. decision tree) will take as input the instance vectors after having been processed by both feature subspacing and feature dimensionality reduction as shown in Algorithm 1.

3.3.2. Ensemble of Kernel Ridge Regression Learners (EnsemKRR)

The base learner for *EnsemKRR* is *RLS-avg*, and *EnsemKRR* does not have the first step of undersampling the majority class as this step is not required for similarity-based methods (such as RLS-avg). Next, we briefly introduce the *RLS-avg* model which was proposed in [15], followed by an overview of *EnsemKRR*.

The first step in *RLS-avg* is to compute kernel matrices for the drugs and targets from D and T, respectively. For example, for a pair of drugs d_i and d_r, the drug similarity matrix K^d is computed as

$$K^d(d_i, d_r) = \exp\left(-\frac{\|D_{i,*} - D_{r,*}\|^2}{p}\right) \quad (1)$$

where $D_{i,*}$ and $D_{r,*}$ are the feature vectors for drugs d_i and d_r, respectively, and p is the number of drug features in D. The target similarity matrix K^t is similarly obtained. Furthermore, two more *Gaussian Interaction Profile* (*GIP*) kernels for drugs and targets are computed from the drug-target interaction matrix Y. Specifically, for a pair of drugs d_i and d_r, the drug GIP kernel K^d_{GIP} is computed as

$$K^d_{GIP}(d_i, d_r) = \exp\left(\frac{\|Y(d_i, *) - Y(d_r, *)\|^2}{\gamma}\right) \quad (2)$$

where $Y(d_i, *)$ and $Y(d_r, *)$ are the rows in the interaction matrix Y that correspond to the interaction profiles of drug d_i and d_r, respectively, and

$$\gamma = \frac{1}{n}\sum_{i=1}^{n}|Y(d_i, *)| \quad (3)$$

where n is the number of drugs. The target GIP kernel K^t_{GIP} is similarly obtained. The GIP kernel K^d_{GIP} is then merged with K^d via linear combination as

$$\widetilde{K}^d = \alpha K^d + (1 - \alpha)K^d_{GIP} \quad (4)$$

where α is an adjustable parameter. $\widetilde{K}^t$ is similarly obtained. The final least-squares solution is then given by

$$\widehat{Y} = \frac{1}{2}\left(\widetilde{K}^d(\widetilde{K}^d + \sigma I)^{-1}Y\right) + \frac{1}{2}\left(\widetilde{K}^t(\widetilde{K}^t + \sigma I)^{-1}Y^\top\right)^\top \quad (5)$$

where σ is a (Tikhonov) regularization parameter.

Based on the above procedure, we can derive the results from the base *RLS-avg* model. Algorithm 2 presents an overview for our proposed *EnsemKRR* method.

Table 3
AUC Results for Feature-based Methods.

Methods	No Dim. Reduction	SVD	PLS	LapEig
DT	0.760	0.746	0.757	0.764
RF	0.855	0.876	0.880	0.874
SVM	0.804	0.818	0.643	0.642
EnsemDT	0.882	0.899	0.902	0.901

Algorithm 2: *EnsemKRR: Kernel Ridge Regression Ensemble*

Input: D = feature matrix for the drugs,
T = feature matrix for the targets,
r = dimensionality reduction parameter,
M = the number of trees in the ensemble

Result: $\widehat{Y}$ = predicted interaction matrix

begin
  for $i \leftarrow 1$ *to* M **do**
    D_i = randomly selected feature subset (of size $r \times p$)
    $D_i = DimRed(D_i)$ //dimensionality reduction
    T_i = randomly selected feature subset (of size $r \times q$)
    $T_i = DimRed(T_i)$ //dimensionality reduction
    $\widehat{Y}_i = RLS\text{-}avg(D_i, T_i)$
  return $\widehat{Y} = \frac{1}{M}\sum_{i=1}^{M}\widehat{Y}_i$

In the procedure of *RLS-avg* described above, each drug is used to train a classifier and each target is used to train a classifier as well, resulting in $p + q$ trained models (where p and q are the number of drugs and targets, respectively). However, when the interaction profile of a drug or target is empty (i.e. it is a *new* drug or target that has no known interaction in Y), this results in a useless model that does not help with prediction. This motivated us to augment the original *RLS-avg* model with a preprocessing procedure called *Weighted Nearest Neighbor* (*WNN*) which was introduced in [27] to infer initial interaction profiles for *new* drugs or targets.

For each new drug d_i, *WNN* infers its interaction profile as

$$\widetilde{Y}(d_i, *) = \sum_{u=1}^{n} w_u Y(d_u, *) \quad (6)$$

where d_1 to d_n are the drugs sorted in descending order based on their similarity to d_i, and $w_i = \eta^{i-1}$ where η is a decay term with $\eta < 1$. Similarly, every new target t has its interaction profile inferred by *WNN* as

$$\widetilde{Y}(*, t_j) = \sum_{v=1}^{m} w_v Y(*, t_v). \quad (7)$$

4. Experimental results and discussion

To demonstrate the prediction performance of *EnsemDT* and *EnsemKRR* as compared with other state-of-the-art methods, we performed a 5-fold cross validation experiment where prediction performance was measured in terms of AUC (*Area Under the ROC Curve*). More precisely, the interaction matrix Y was divided into 5 folds and these folds had turns being left out to act as a test set while the rest were treated as the training set. This procedure led to 5 AUC scores being computed which were then averaged to give the final score. In fact, AUC is insensitive to the high class imbalance present in the data [34], and thus it is widely used for measuring the performance of various prediction methods. Furthermore, we also conducted sensitivity analyses for the parameters of *EnsemDT* and *EnsemKRR*.

Note that Dataset 1 is the main dataset that is used to perform most of the experiments in this paper, while Dataset 2 is only used in Section 4.7.

When dimensionality reduction is used, each fold separately undergoes a dimensionality reduction step *before* running the prediction method, unless the prediction method used is *EnsemDT* or

EnsemKRR. In the case when either *EnsemDT* or *EnsemKRR* is used, dimensionality reduction is done *after* the feature subspacing step in these methods.

4.1. Comparison between the feature-based methods

We compare between the different feature-based methods: *DT*, *RF*, *SVM* and *EnsemDT*. Table 3 shows the comparison among these feature-based methods in terms of AUC. Default parameter values were used for *RF* and *SVM* as defined in MATLAB's *TreeBagger* and *fitcsvm*, respectively. For *EnsemDT*, the parameter values were set as $M = 50$, $npRatio = 5$ and $r = 0.2$.

We first observe that both *RF* and *EnsemDT* have benefited from all dimensionality reduction techniques. This is quite satisfactory since dimensionality reduction also considerably decreased the running time for *RF* and *EnsemDT* (see Section 4.6). The highest AUC results for *EnsemDT* and *RF* were obtained using the *PLS* dimensionality reduction technique, with *SVD* and *Laplacian Eigenmaps* producing results comparable to those of *PLS*. As for *SVM*, it benefited from *SVD*, whereas it did not benefit from either *PLS* or *Laplacian Eigenmaps* as the prediction performance actually degraded. These results suggest that *PLS* and *Laplacian Eigenmaps* are not a good fit with *SVM*. In the end, *EnsemDT* is the best out of the four feature-based methods.

4.2. Varying training sets for EnsemDT's base learners

As shown in Table 3, *EnsemDT* produced better results than *RF*. In fact, there are two main differences between *EnsemDT* and *RF*. First, *EnsemDT* randomly samples a different negative set for each base learner. This is unlike *RF* which performs bagging on the same negative set. Second, *EnsemDT* includes all of the positive instances in every base learner, while *RF* performs bagging on the positive instances along with the negative instances. That is, each decision tree in *RF* uses only a subset of the positive instances.

We further created different variants of *EnsemDT* by using different samples to train the base learners. The first variant randomly samples a negative set (of instances), merges it with the positive set to form a training set, and uses that same training set in all base learners of the ensemble. The second variant also uses the same training set, but additionally does bagging on it for each base learner. The third variant is similar to the second variant, but instead of bagging on this training set entirely, bagging is performed only on its negative instances while retaining all the positive instances. Finally, the fourth variant is the originally proposed variant of *EnsemDT* that is included in Table 3. Like the third variant, it uses all the positive instances in every base learner but, instead of bagging on the same negative instances, samples a different negative set for each base learner. The results of the different variants are given in Table 4. Note that no dimensionality reduction was employed in these variants as we wanted the results to be free from any influence that dimensionality reduction may have on them. This way, the conclusions drawn would be more reliable.

The first variant (i.e. same training set in all learners) provides us with a baseline against which to compare the other variants. The second variant (i.e. bagging on the same training set for all learners) produced a result that is lower than the first variant, which is a surprising observation since bagging is a mechanism that is typically employed in ensembles to improve the prediction performance [22]. However, we can identify the cause of this observation by looking at the result of the third variant (i.e. same training set with bagging only on negative instances). By avoiding bagging on the positive instances, the result improved as compared to that of the second variant. As such, we conclude that the prediction performance is very sensitive to any positive instances being left out. That is, the best strategy is to include all positive instances in all base learners of EnsemDT. The last variant (i.e. all positive instances and different negative sets for the learners) gives the best result out of all the variants as it incorporates more data into the training of the ensemble.

Table 4
AUC Results for the Variants of *EnsemDT*.

Methods	AUC
Same Training Set	0.874
Same Training Set + Bagging	0.867
Entire Positive Set + Bagging on Negatives only	0.876
Entire Positive Set + Different Negative Sets	0.882

4.3. Comparison between the similarity-based methods

We then compare between the different similarity-based methods: *NP*, *WP*, *NBI*, *RLS-avg*, *RLS-kron* and *EnsemKRR*. Table 5 contains the AUC scores for various similarity-based methods. *NP*, *WP* and *NBI* do not have any parameters to tune. As for *RLS-avg* and *RLS-kron*, default values for α and σ are used as specified in [15], while η was set to 0.7 as it was determined in [19] to be a good default value. For *EnsemKRR*, the parameter values were set as $M = 50$ and $r = 0.2$.

Clearly, *RLS-avg* and *RLS-kron* performed much better than *NP*, *WP* and *NBI*. As an ensemble made of *RLS-avg* learners, *EnsemKRR* produced better results than all the other methods. By simply performing feature subspacing for each base learner, it was able to produce a satisfactory AUC result of 94.3%.

An important observation here is that similarity-based methods generally do not benefit from dimensionality reduction. This is attributed to the way the input data is presented to these methods, which is in the form of similarity or kernel matrices for the drugs and targets. Dimensionality reduction techniques retain the similarity between instances in the reduced feature space. Therefore, the similarity matrices before and after dimensionality reduction are expected to be similar, leading to no improvement for these similarity-based methods after dimensionality reduction.

4.4. Effect of GIP kernels and WNN

We noticed from the results in Table 5 that *NP*, *WP* and *NBI* are not doing as well as *RLS-avg* and *RLS-kron*. One of the potential factors that may be leading to this performance gap is that *NP*, *WP* and *NBI* use drug and target kernels K^d and K^t (obtained from Eq. (1)), whereas *RLS-avg* and *RLS-kron* additionally make use of GIP kernels (see Eq. (2)) to merge with K^d and K^t prior to prediction. As such, we studied the effect of GIP kernels on all similarity-based methods. Table 6 shows the results for the different methods with and without the GIP kernels (no dimensionality reduction applied). Note that when GIP kernels are used, WNN is also applied to Y beforehand.

It is obvious from the results of all the methods that GIP kernels and WNN have a large positive impact on the prediction perfor-

Table 5
AUC Results for Similarity-based Methods

Methods	No Dim. Reduction	SVD	PLS	LapEig
NP	0.679	0.679	0.670	0.626
WP	0.793	0.793	0.792	0.791
NBI	0.606	0.601	0.613	0.618
RLS-avg	0.925	0.912	0.915	0.909
RLS-kron	0.915	0.900	0.904	0.894
EnsemKRR	0.943	0.942	0.941	0.941

Table 6
AUC Results for Interaction Prediction with and without WNN & GIP.

Methods	No WNN, No GIP	WNN & GIP
NP	0.679	0.690
WP	0.793	0.878
NBI	0.606	0.870
RLS-avg	0.873	0.925
RLS-kron	0.786	0.915
EnsemKRR	0.866	0.943

Table 7
AUC Results for Interaction Prediction with and without WNN and/or GIP.

Methods	No GIP, No WNN	WNN	GIP	WNN & GIP
RLS-avg	0.873	0.908	0.795	0.925
EnsemKRR	0.866	0.919	0.771	0.943

mance. For example, *WP* and *NBI* now have results that are comparable to those of feature-based methods from Table 3. An interesting observation here is *RLS-avg* is better than *EnsemKRR* when neither WNN nor GIP is used. It seems that feature subspacing did not help improve the prediction performance in this case. It is possible that the value of r (portion of randomly selected features per base learner) simply needs to be adjusted.

We further compared *RLS-avg* and *EnsemKRR* with and without the use of WNN or GIP kernels as shown in Table 7.

Results in Table 7 show that prediction performance improves when WNN is applied prior to prediction whether or not GIP kernels are being computed. However, when the computation of the GIP kernels is attempted without having applied WNN beforehand, prediction performance is seriously affected. This is due to the presence of *empty* interaction profiles in Y; that is, there are *new* drugs and targets that have no known interactions in Y. As mentioned in Section 3.3.2, creating GIP kernels involves computing similarities between interaction profiles of pairs of drugs or pairs of targets. The problem is that when some of these interaction profiles are empty, the resulting computed similarity can be very misleading because these empty profiles will be incorrectly considered identical, while in reality they are cases of missing information. The presence of such cases leads to improper GIP kernels that worsen the prediction performance.

4.5. Sensitivity analysis

Here, we provide sensitivity analysis for two parameters M (number of base learners) and r (portion of features to randomly select per base learner) under the two proposed methods *EnsemDT* and *EnsemKRR*. Figs. 1 and 2 contain the sensitivity analysis for M and r, respectively. Regarding M, the AUC for *EnsemDT* seems to improve quickly from $M = 1$ till $M = 50$, after which it still steadily improves but at a slower rate. As for *EnsemKRR*, it seems not to improve much beyond $M = 20$. Regarding r, *EnsemDT* seems to be quite robust to the value of r till $r = 0.8$, after which the AUC decreases. As for *EnsemKRR*, it shows a steady decrease in AUC as the value of r increases, meaning that it performs best at $0.1 \leqslant r \leqslant 0.2$.

4.6. Running times

The cross validation experiments have been performed on an Intel Xeon CPU (E5-1620 0 @ 3.60 GHz). For feature-based methods, dimensionality reduction helped to significantly decrease the running time. The running time for all feature-based methods are shown in Table 8 below.

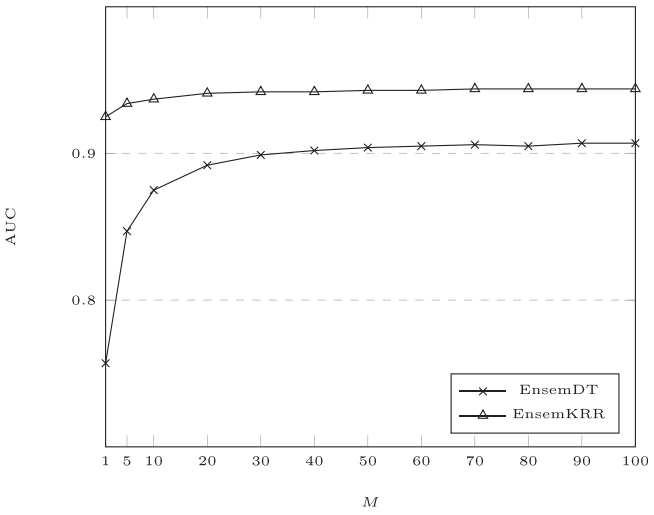

Fig. 1. Sensitivity analysis for M.

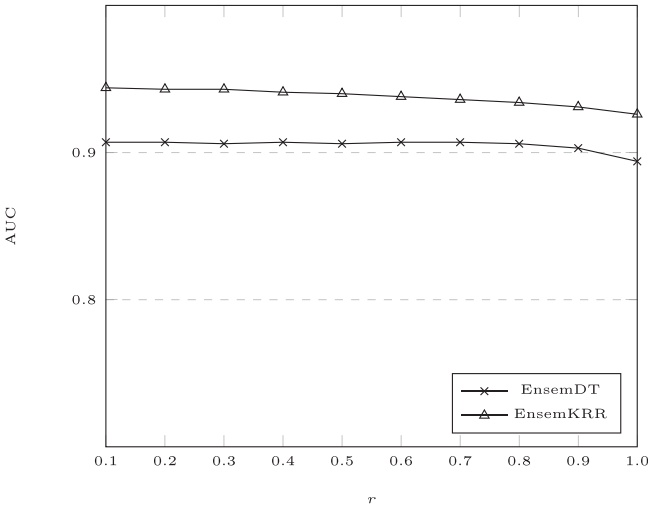

Fig. 2. Sensitivity analysis for r.

As shown in Table 8, dimensionality reduction greatly decreases the running times. *SVM* in particular was suffering from the high dimensionality of the full feature set. Note that for *EnsemDT*, the dimensionality reduction is done for every base learner (on its subspaced feature set), while it is done only once prior to prediction for the rest of the methods. This explains why the running time improvement for *EnsemDT* is not as strong as that for, say, *RF* when each is compared to its own running time without dimensionality reduction. For example, when SVD is used, *EnsemDT* gains a speedup of 3.31, while *RF* gains a higher speedup of 4.85.

An additional note that we wish to mention is that *RLS-avg* and, by extension, *EnsemKRR* run reasonably fast. Without dimensionality reduction their running times are 2 and 68 min, respectively. We believe that this is a byproduct of its base learners being bipartite local models that separate the prediction problem into drug and target sides (i.e. the main prediction problem is divided into two smaller sub-problems) that are solved separately and then have their results merged to give the final result.

4.7. Experimenting with Dataset 2

Here, we present experimental results for running the prediction methods covered in this paper on Dataset 2 (i.e. the one that

Table 8
Running Time for Various Feature-based Methods (mins).

Methods	No Dim. Reduction	SVD	PLS	LapEig
DT	5	1	4	4
RF	921	190	208	211
SVM	2,636	84	113	117
EnsemDT	86	26	43	163

Table 9
AUC Results using Dataset 2.

Methods	No Dim. Reduction	SVD	PLS	LapEig
DT	0.808	0.781	0.801	0.791
RF	0.762	0.891	0.891	0.893
SVM	0.733	0.733	0.697	0.627
EnsemDT	0.886	0.914	0.898	0.914
NP	0.700	0.638	0.682	0.628
WP	0.822	0.868	0.818	0.817
NBI	0.582	0.579	0.687	0.632
RLS-avg	0.928	0.899	0.918	0.916
RLS-kron	0.909	0.873	0.913	0.874
EnsemKRR	0.933	0.931	0.930	0.930

is presented in [30]). For details on this dataset, the reader is referred back to Section 3 of this paper. The results of this experiment are given in Table 9.

Before commenting on the results, note that the value of the r parameter (i.e. the portion of randomly selected features per base learner) was altered to optimize the prediction performance for *EnsemDT* on this dataset; the new value is $r = 0.9$ (i.e. feature subspacing is almost turned off). There is a characteristic in this dataset that forced this modification of the r parameter, namely the very sparse nature of the domain fingerprints representing the targets. Indeed, the average number of domains per target is 1.51 (out of 876 domains). Due to that sparse nature of the target feature vectors, feature subspacing was decreased to avoid the loss of what little information there is available for each target.

Interestingly, in contrast to *EnsemDT*, the optimal value of r when using *EnsemKRR* on this dataset is $r = 0.1$. In *RLS-avg*, the base learner used in *EnsemKRR*, the similarity matrices for the drugs and targets are first generated from their feature vectors. As a result of the target feature vectors being very sparse, the target similarity matrix is not expected to change much with feature subspacing applied. As for the drug similarity matrix, doing feature subspacing on the drug features beforehand would help inject diversity into the ensemble, thus improving prediction performance. This is why the optimal value is as low as $r = 0.1$ for *EnsemKRR*.

After having adjusted the value of r for optimal performance and looking at the results in Table 9, the conclusions are almost the same as those drawn from the results in Tables 3 and 5. However, there are two main differences.

The first difference is that *RF* produces a lower AUC than *DT* when dimensionality reduction is not performed (the first column of Table 9). This can be explained by the fact that the target feature vectors are too sparse and, as such, the prediction performance is hurt by the feature subspacing that is available in *RF* by default. However, this phenomenon does not happen when dimensionality reduction is used because, in case of *RF*, dimensionality reduction is done *before* running the *RF* algorithm on the reduced (and, thus, non-sparse) features.

The other difference is that *PLS* is no longer the superior dimensionality reduction technique for use with *EnsemDT*. *PLS* is generally suited for cases where there is multicollinearity in the features [35], which is not the case in the sparse domain fingerprints of the targets.

In the end, we conclude by saying that the conclusions that were drawn earlier in this paper are generalizable to most datasets. Whatever differences in results that were seen on Dataset 2 (as opposed to Dataset 1) were consequences of a characteristic that is unique to it, namely the excessive sparseness of the target feature vectors.

4.8. Discussions

A simple comparison between feature-based methods and similarity-based methods shows that similarity-based methods are more promising. *WP* and *NBI* (after inclusion of GIP kernels) produced a result that easily exceeds that of *SVM* and that is comparable to that of *RF*. Stand-alone *RLS-avg* produced results that are superior to those of the best feature-based method, *EnsemDT*. After using *RLS-avg* as a base learner in an ensemble, its results improved even further.

We believe the good performance of *RLS-avg* is due to two reasons. The first is that it is a bipartite local model, which relieves it from having to figure out a suitable way to merge the drug and target information in order to form feature vectors for the instances (as opposed to concatenating drug and target feature vectors to form instances in feature-based methods). Secondly, they make good use of an extra source of information, the network similarity derived from the interaction matrix Y (in the form of GIP kernels), by merging it with the drug and target similarity matrices that were derived from the initial drug and target features to assist with prediction.

In addition, there was the subtle addition to *RLS-avg* that led to the prediction performance that was reached in our experiments; that is, the applying of the WNN procedure [27] to the interaction matrix Y prior to prediction.

5. Conclusion

In this work, we presented two ensemble learning methods, namely *EnsemDT* and *EnsemKRR*, for predicting drug-target interactions. Particularly, *EnsemDT* is a feature-based method that employs *Decision Tree* as its base learner, while *EnsemKRR* is a similarity-based method that employs *RLS-avg* [15] as its base learner. Based on standard cross validation experiments, *EnsemDT* was compared with other feature-based state-of-the-art methods, and *EnsemKRR* was compared with other similarity-based state-of-the-art methods. Our experimental results demonstrate that both methods have performed very well compared with existing methods. Nevertheless, we observe that while each of *EnsemDT* and *EnsemKRR* was superior in its own category, the experiments have shown that similarity-based methods are more promising than feature-based methods. This is likely due to the data representation in the feature-based methods that concatenates drug and target feature vectors to form unified instance feature vectors. This is an issue that similarity-based methods avoid via the concept of bipartite local models that divides the prediction problem into two independent sub-problems (one from the drug side, the other from the target side). It solves each of the sub-problems separately and then merges their results to give the final scores.

We also proposed to use dimensionality reduction techniques to make the prediction problem more manageable in terms of running time and space complexity. This helped the feature-based methods run more efficiently with the additional bonus of improved prediction performance. However, the similarity-based methods did not benefit from dimensionality reduction, which is possibly due to the way that data is given as input to these methods. That is, reducing the dimensionality of the drug and target features before computing kernels from them did not help improve

the prediction performance. On the contrary, the prediction performance decreased in most cases where dimensionality reduction was used with similarity-based methods.

We made use of the WNN procedure [27] to further enhance the prediction performance of *RLS-avg* before including it as a base learner in *EnsemKRR*. Indeed, its inclusion led to significant improvements, and it was found to be quite important when considering to make use of generated GIP kernels (i.e. network similarity matrices) from the data.

Experiments of *EnsemDT* have also shown an interesting observation. That is, performing bagging on the positive instances was found to degrade the prediction performance, which suggests that leaving out some of the positive instances per base learner is not a good idea, and all the positive examples should be fully leveraged for building the classification model. It may be possible that the *unlabeled* instances (that we treat as negative instances throughout the paper) actually contain some undiscovered positive instances. We can further improve this work by extracting those likely positives that can be integrated with original known positives to better represent the positive class. In addition, by doing so, we could also reduce the false negatives in the unlabeled instances to get more purer negative data. We leave this as future work.

References

[1] J. Li, S. Zheng, B. Chen, A.J. Butte, S.J. Swamidass, Z. Lu, A survey of current trends in computational drug repositioning, Briefings Bioinf. 17 (1) (2016) 2–12, http://dx.doi.org/10.1093/bib/bbv020.

[2] S. Fakhraei, B. Huang, L. Raschid, L. Getoor, Network-based drug-target interaction prediction with probabilistic soft logic, IEEE/ACM Trans. Comput. Biol. Bioinf. 11 (5) (2014) 775–787, http://dx.doi.org/10.1109/TCBB.2014.2325031.

[3] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, M. Tanabe, KEGG for integration and interpretation of large-scale molecular data sets, Nucl. Acids Res. 40 (D1) (2012) D109–D114, http://dx.doi.org/10.1093/nar/gkr988.

[4] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, A. Pon, K. Banco, C. Mak, V. Neveu, Y. Djoumbou, R. Eisner, A.C. Guo, D.S. Wishart, DrugBank 3.0: a comprehensive resource for 'omics' research on drugs, Nucl. Acids Res. 39 (suppl 1) (2011) D1035–D1041, http://dx.doi.org/10.1093/nar/gkq1126.

[5] A. Gaulton, L.J. Bellis, A.P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, J.P. Overington, ChEMBL: a large-scale bioactivity database for drug discovery, Nucl. Acids Res. 40 (D1) (2011) D1100–D1107, http://dx.doi.org/10.1093/nar/gkr777.

[6] M. Kuhn, D. Szklarczyk, S. Pletscher-Frankild, T.H. Blicher, C. von Mering, L.J. Jensen, P. Bork, STITCH 4: integration of protein chemical interactions with user data, Nucl. Acids Res. 42 (D1) (2014) D401–D407, http://dx.doi.org/10.1093/nar/gkt1207.

[7] G. Jin, S.T. Wong, Toward better drug repositioning: prioritizing and integrating existing methods into efficient pipelines, Drug Discovery Today 19 (5) (2014) 637–644, http://dx.doi.org/10.1016/j.drudis.2013.11.005.

[8] H. Li, Z. Gao, L. Kang, H. Zhang, K. Yang, K. Yu, X. Luo, W. Zhu, K. Chen, J. Shen, X. Wang, H. Jiang, TarFisDock: a web server for identifying drug targets with docking approach, Nucl. Acids Res. 34 (suppl 2) (2006) W219–W224, http://dx.doi.org/10.1093/nar/gkl114.

[9] L. Xie, T. Evangelidis, L. Xie, P.E. Bourne, Drug discovery using chemical systems biology: weak inhibition of multiple kinases may contribute to the anti-cancer effect of nelfinavir, PLoS Comput. Biol. 7 (4) (2011) 1–13, http://dx.doi.org/10.1371/journal.pcbi.1002037.

[10] Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, M. Kanehisa, Prediction of drug–target interaction networks from the integration of chemical and genomic spaces, Bioinformatics 24 (13) (2008) i232–i240, http://dx.doi.org/10.1093/bioinformatics/btn162.

[11] H. Yu, J. Chen, X. Xu, Y. Li, H. Zhao, Y. Fang, X. Li, W. Zhou, W. Wang, Y. Wang, A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data, PLoS ONE 7 (5) (2012) 1–14, http://dx.doi.org/10.1371/journal.pone.0037608.

[12] Z. He, J. Zhang, X.-H. Shi, L.-L. Hu, X. Kong, Y.-D. Cai, K.-C. Chou, Predicting drug-target interaction networks based on functional groups and biological features, PloS one 5 (3) (2010) e9603, http://dx.doi.org/10.1371/journal.pone.0009603.

[13] A. Ezzat, M. Wu, X.-L. Li, C.-K. Kwoh, Drug-target interaction prediction via class imbalance-aware ensemble learning, BMC Bioinf. 17 (19) (2016) 267–276, http://dx.doi.org/10.1186/s12859-016-1377-y.

[14] K. Bleakley, Y. Yamanishi, Supervised prediction of drug–target interactions using bipartite local models, Bioinformatics 25 (18) (2009) 2397–2403, http://dx.doi.org/10.1093/bioinformatics/btp433.

[15] T. van Laarhoven, S.B. Nabuurs, E. Marchiori, Gaussian interaction profile kernels for predicting drug–target interaction, Bioinformatics 27 (21) (2011) 3036–3043, http://dx.doi.org/10.1093/bioinformatics/btr500.

[16] J.-P. Mei, C.-K. Kwoh, P. Yang, X.-L. Li, J. Zheng, Drug-target interaction prediction by learning from local information and neighbors, Bioinformatics 29 (2) (2013) 238–245, http://dx.doi.org/10.1093/bioinformatics/bts670.

[17] M. Gönen, Predicting drug–target interactions from chemical and genomic kernels using bayesian matrix factorization, Bioinformatics 28 (18) (2012) 2304–2310, http://dx.doi.org/10.1093/bioinformatics/bts360.

[18] X. Zheng, H. Ding, H. Mamitsuka, S. Zhu, Collaborative matrix factorization with multiple similarities for predicting drug-target interactions, Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2013, pp. 1025–1033, http://dx.doi.org/10.1145/2487575.2487670.

[19] A. Ezzat, P. Zhao, M. Wu, X. Li, C.K. Kwoh, Drug-target interaction prediction with graph regularized matrix factorization, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2016, p. 99, http://dx.doi.org/10.1109/TCBB.2016.2530062. 1–1.

[20] X. Chen, M.-X. Liu, G.-Y. Yan, Drug-target interaction prediction by random walk on the heterogeneous network, Mol. BioSyst. 8 (2012) 1970–1978, http://dx.doi.org/10.1039/C2MB00002D.

[21] F. Cheng, C. Liu, J. Jiang, W. Lu, W. Li, G. Liu, W. Zhou, J. Huang, Y. Tang, Prediction of drug-target interactions and drug repositioning via network-based inference, PLoS Comput. Biol. 8 (5) (2012) e1002503, http://dx.doi.org/10.1371/journal.pcbi.1002503.

[22] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, CRC Press, 2012.

[23] S.Y. Kung, Kernel Methods and Machine Learning, Cambridge University Press, 2014.

[24] Z. Mousavian, A. Masoudi-Nejad, Drug-target interaction prediction via chemogenomic space: learning-based methods, Expert Opin. Drug Metabol. Toxicol. 10 (9) (2014) 1273–1287, http://dx.doi.org/10.1517/17425255.2014.950222.

[25] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, http://dx.doi.org/10.1023/A:1010933404324.

[26] N. Cristianini, J. Shawe-Taylor, Support vector machines and other kernel-based learning methods, 2000.

[27] T. van Laarhoven, E. Marchiori, Predicting drug-target interactions for new drug compounds using a weighted nearest neighbor profile, PloS One 8 (6) (2013) e66952.

[28] D.-S. Cao, N. Xiao, Q.-S. Xu, A.F. Chen, Rcpi: R/bioconductor package to generate various descriptors of proteins, compounds and their interactions, Bioinformatics 31 (2) (2015) 279–281, http://dx.doi.org/10.1093/bioinformatics/btu624.

[29] Z.R. Li, H.H. Lin, L.Y. Han, L. Jiang, X. Chen, Y.Z. Chen, PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence, Nucl. Acids Res. 34 (suppl 2) (2006) W32–W37, http://dx.doi.org/10.1093/nar/gkl305.

[30] Y. Tabei, E. Pauwels, V. Stoven, K. Takemoto, Y. Yamanishi, Identification of chemogenomic features from drug–target interaction networks using interpretable classifiers, Bioinformatics 28 (18) (2012) i487–i494, http://dx.doi.org/10.1093/bioinformatics/bts412.

[31] R.D. Finn, P. Coggill, R.Y. Eberhardt, S.R. Eddy, J. Mistry, A.L. Mitchell, S.C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas, et al., The Pfam protein families database: towards a more sustainable future, Nucl. Acids Res. 44 (D1) (2016) D279–D285, http://dx.doi.org/10.1093/nar/gkv1344.

[32] S. de Jong, Simpls: an alternative approach to partial least squares regression, Chemom. Intell. Lab. Syst. 18 (3) (1993) 251–263, http://dx.doi.org/10.1016/0169-7439(93)85002-X.

[33] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: T. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems 14, MIT Press, 2002, pp. 585–591.

[34] T. Fawcett, An introduction to ROC analysis, Pattern Recogn. Lett. 27 (8) (2006) 861–874, http://dx.doi.org/10.1016/j.patrec.2005.10.010.

[35] O. Erdas, E. Buyukbingol, F.N. Alpaslan, A. Adejare, Modeling and predicting binding affinity of phencyclidine-like compounds using machine learning methods, J. Chemom. 24 (1) (2010) 1–13, http://dx.doi.org/10.1002/cem.1265.